REMARKS

In accordance with the foregoing, claims 1, 5, and 9-10 have been amended, and claims 1-14 are pending and under consideration. No new matter is presented in this Amendment.

REJECTIONS UNDER 35 U.S.C. §102:

Claims 1-4 and 13-14 are rejected under 35 U.S.C. §102(e) as being anticipated by Stone et al. (U.S. Patent 6,504,554), hereinafter "Stone." The Applicants respectfully traverse the rejection and request reconsideration.

Regarding the rejection of independent claim 1, it is noted that claim 1 recites "[a] method of executing markup document **applet** by a browser, comprising... loading the requested applet into a virtual machine." In contrast, Stone only discloses a method of converting a Java **object** to a tag-based code (such as HTML). While the Examiner cites the objects of Stone as a teaching of the applet in the present claim, the Applicants respectfully submit that an applet and a Java object are patentably distinct. Specifically, an applet is a program executable by a virtual machine (such as the Java Virtual Machine). However, a Java object is a component of a Java application. In Stone, for example, it is explained that a "programmer creates and manipulates objects that correspond to components of a web page" (column 2, lines 1-2). In turn, the runtime environment disclosed in Stone converts the Java object into a tag-based display language (column 5, lines 27-29). Thus, the runtime environment of Stone is different from the virtual machine recited in the present claim, which loads an applet (as opposed to converting an object). Similarly, the "bound applet" and "unbound applet" of the present claim are different from the "bound object" (a Java object that has been converted to an HTML element and modified or had an event attached thereto) and "unbound object" (a Java object) of Stone. While the present claim relates to the executing of an applet, Stone relates to the converting or translating – and not executing or loading - of a Java object into HTML. Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, a method of executing an applet, as recited in claim 1.

Regarding the rejection of claims 2 and 3, it is noted that these claims depend from claim 1 and are, therefore, allowable for at least the reasons set forth above.

Regarding the rejection of claim 4, it is noted that this claim depends from claim 3 and is, therefore, allowable for at least the reasons set forth above.

Regarding the rejection of independent claim 13, it is noted that claim 13 recites a method of "classifying tagged **applets** of a markup document, and controlling different execution

life cycles of the tagged applets according to the classifying." In contrast, Stone only discloses a method of converting a Java **object** to a tag-based code (such as HTML). While the Examiner cites the objects of Stone as a teaching of the applet in the present claim, the Applicants respectfully submit that an applet and a Java object are patentably distinct. Specifically, an applet is a program executable by a virtual machine (such as the Java Virtual Machine). However, a Java object is a component of a Java application. In Stone, for example, it is explained that a "programmer creates and manipulates objects that correspond to components of a web page" (column 2, lines 1-2). While the present claim relates to the classifying of an applet, Stone relates to the converting or translating – and not classifying or executing - of a Java object into HTML. Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, a method of classifying an applet and controlling a life cycle thereof according to the classifying, as recited in claim 13.

Regarding the rejection of claim 14, it is noted that this claim depends from claim 13 and is, therefore, allowable for at least the reasons set forth above. Furthermore, it is noted that claim 14 recites classifying the applet "into bound and unbound applets, and wherein... the execution life cycle of an unbound applet is independent of the markup document life." In contrast, Stone discloses a "bound object" and an "unbound object" (column 7, lines 30-48). The "bound applet" and "unbound applet" of the present claim are different from the "bound object" (a Java object that has been converted to an HTML element and modified or had an event attached thereto) and "unbound object" (a Java object) of Stone. For example, while an unbound applet as taught by the present claim has an execution life cycle, the unbound object of Stone is not executed, but in a state prior to (or after) conversion to HTML (column 7, lines 30-48). Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, a method of classifying an applet as bound or unbound, as recited in claim 14.

REJECTIONS UNDER 35 U.S.C. §103:

Claims 5-12 are rejected under 35 U.S.C. §103(a) as being unpatentable over Stone in view of Engstrom et al. (U.S. Publication 2005/0120305), hereinafter "Engstrom." The Applicants respectfully traverse the rejection and request reconsideration.

Regarding the rejection of independent claim 5, it is noted that claim 5 recites an application manager that receives and loads an **applet** from an external data source into a virtual machine and "determines whether the loaded applet is a bound applet or an unbound applet." In contrast, Stone only discloses a method of converting a Java **object** to a tag-based

code (such as HTML). While the Examiner cites the objects of Stone as a teaching of the applet in the present claim, the Applicants respectfully submit that an applet and a Java object are patentably distinct. Specifically, an applet is a program executable by a virtual machine (such as the Java Virtual Machine). However, a Java object is a component of a Java application. In Stone, for example, it is explained that a "programmer creates and manipulates objects that correspond to components of a web page" (column 2, lines 1-2). In turn, the runtime environment disclosed in Stone converts the Java object into a tag-based display language (column 5, lines 27-29). Thus, the runtime environment of Stone is different from the virtual machine recited in the present claim, which loads an applet (as opposed to converting an object). Similarly, the "bound applet" and "unbound applet" of the present claim are different from the "bound object" (a Java object that has been converted to an HTML element and modified or had an event attached thereto) and "unbound object" (a Java object) of Stone. While the present claim relates to the executing of an applet, Stone relates to the converting or translating – and not executing or loading - of a Java object into HTML. Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, an apparatus to execute an applet, as recited in claim 5.

Regarding the rejection of claims 6 and 7, it is noted that these claims depend from claim 5 and are, therefore, allowable for at least the reasons set forth above.

Regarding the rejection of claim 8, it is noted that this claim depends from claim 7 and is, therefore, allowable for at least the reasons set forth above.

Regarding the rejection of claim 9, it is noted that claim 9 recites a determining of whether a "requested **applet** is a bound applet or an unbound applet" and "loading the requested applet into a virtual machine." In contrast, Stone only discloses a method of converting a Java **object** to a tag-based code (such as HTML). While the Examiner cites the objects of Stone as a teaching of the applet in the present claim, the Applicants respectfully submit that an applet and a Java object are patentably distinct. Specifically, an applet is a program executable by a virtual machine (such as the Java Virtual Machine). However, a Java object is a component of a Java application. In Stone, for example, it is explained that a "programmer creates and manipulates objects that correspond to components of a web page" (column 2, lines 1-2). In turn, the runtime environment disclosed in Stone converts the Java object into a tag-based display language (column 5, lines 27-29). Thus, the runtime environment of Stone is different from the virtual machine recited in the present claim, which loads an applet (as opposed to converting an object). Similarly, the "bound applet" and "unbound applet" of the

present claim are different from the "bound object" (a Java object that has been converted to an HTML element and modified or had an event attached thereto) and "unbound object" (a Java object) of Stone. While the present claim relates to the loading of an applet, Stone relates to the converting or translating – and not executing or loading - of a Java object into HTML. Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, a method of loading an applet, as recited in claim 9.

Regarding the rejection of independent claim 10, it is noted that claim 10 recites a determining of "whether [a] loaded **applet** is a bound applet or an unbound applet." In contrast, Stone only discloses a method of converting a Java **object** to a tag-based code (such as HTML). While the Examiner cites the objects of Stone as a teaching of the applet in the present claim, the Applicants respectfully submit that an applet and a Java object are patentably distinct. Specifically, an applet is a program executable by a virtual machine (such as the Java Virtual Machine). However, a Java object is a component of a Java application. In Stone, for example, it is explained that a "programmer creates and manipulates objects that correspond to components of a web page" (column 2, lines 1-2). In turn, the runtime environment disclosed in Stone converts the Java object into a tag-based display language (column 5, lines 27-29). Thus, the runtime environment of Stone is different from the virtual machine recited in the present claim, which loads an applet (as opposed to converting an object). Similarly, the "bound applet" and "unbound applet" of the present claim are different from the "bound object" (a Java object that has been converted to an HTML element and modified or had an event attached thereto) and "unbound object" (a Java object) of Stone. While the present claim relates to the launching of an applet, Stone relates to the converting or translating – and not executing or loading - of a Java object into HTML. Therefore, the Applicants respectfully submit that Stone fails to disclose, implicitly or explicitly, a method of launching an applet, as recited in claim 10.

Regarding the rejection of claims 11 and 12, it is noted that these claims depend from claim 10 and are, therefore, allowable for at least the reasons set forth above.

Based on the foregoing, this rejection is respectfully requested to be withdrawn.

CONCLUSION:

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is

requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 503333.

Respectfully submitted,

STEIN, MCEWEN & BUI, LLP

Date: 11/2/07

By: _____
Michael D. Stein
Registration No. 37,240

1400 Eye St., NW
Suite 300
Washington, D.C. 20005
Telephone: (202) 216-9505
Facsimile: (202) 216-9510